GENERAL INSTRUCTIONS FOR

ISMLR APPLICATION

VERSION 1.2

Junjie Zhu

Illinois Institute of Technology

Initial published date: August 1, 2017

Last revised: March 22, 2018

UPDATES

Version	Published date	Major updates and revisions
1.0	08/01/2017	
1.1	01/18/2018	• Combined 1st regressor and 2nd regressor together to avoid confusion • Changed time format “yyyy/mm/dd HH:MM” to a format depends on the input information on GUI and in excels • Revised scripts to show the name of response regressor in GUI plots and excels • Downsized fonts in GUI plots
1.2	03/22/2018	• Included “Future information prediction tool”

Table of Content

INTRODUCTION

This document is a general guide describing use of the *ISMLR application version 1.2*. More information about conventional ISMLR method can be found in the description by Zhu and Anderson (2016). A use of ISMLR method that integrates with a decision-making technique is described in Zhu et al. (2018). This application guide was developed to facilitate a more efficient and wider use of the ISMLR method. The ISMLR application is designed to process a large time-series dataset to:

- Provide a primary prediction of the response variable
- Minimize the amount of missing data from a regression analysis
- Select a subset of important regressors, or
- Produce pretreated datasets for a subsequent prediction using a more advanced algorithm

At the current version, a future information prediction tool is included, so values of future response variable can be predicted at one modeling action.

Although the ISMLR application is easy to understand and use, reading this document before running the program will help you to better implement the application.

APPLICATION REQUIREMENT

To run the ISMLR application you need to have MATLAB Compiler Runtime (MCR) installed. Any computer (64-bit operating system) that has MATLAB version R2016b installed can directly run the executable file "*ISMLR application.exe*" (file size < 20 MB). To run this file on computer that has an earlier version of MATLAB or does not have MATLAB installed, requires installation of MCR, version 2016b. You can download the MCR from their website: https://www.mathworks.com/products/compiler/mcr.html

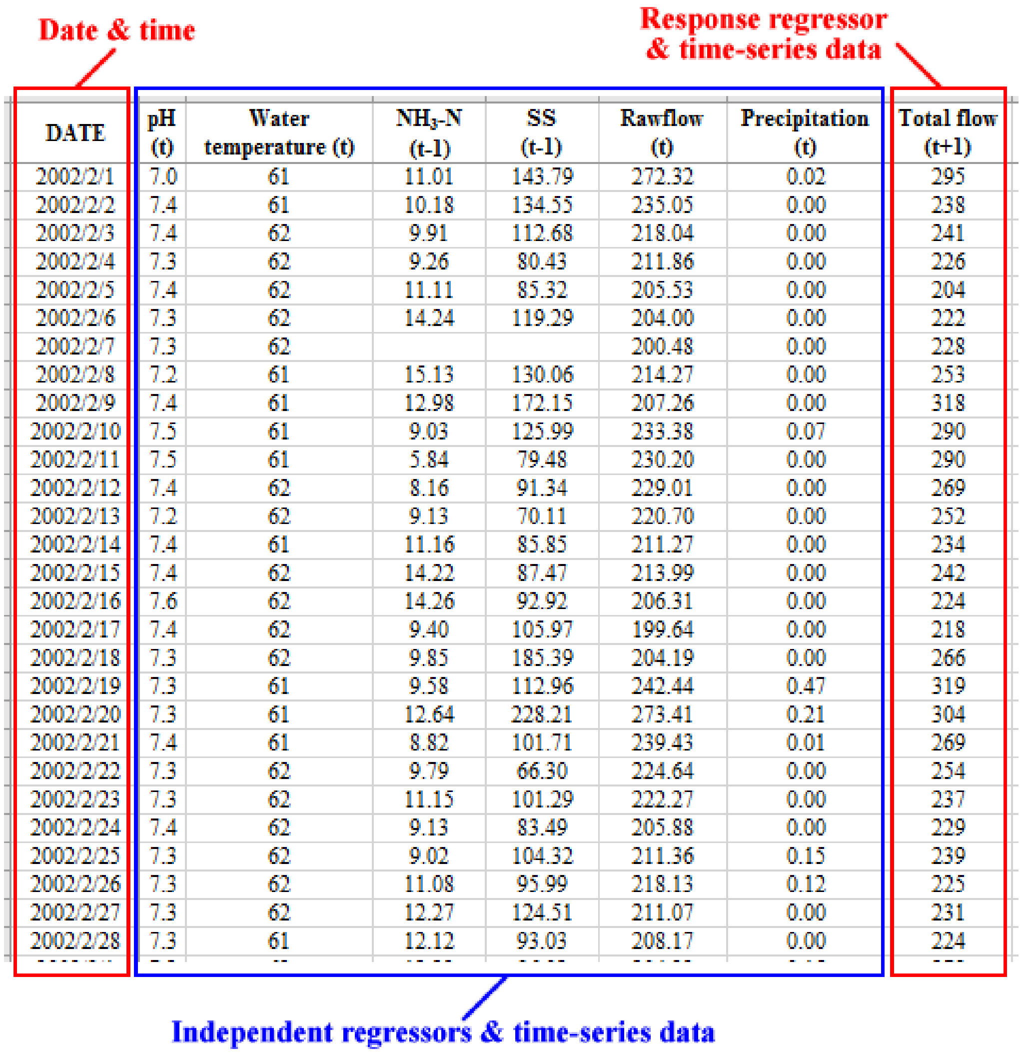
Summary of requirement

- Windows, 64-bit operating system
- MATLAB, version 2016b; or MCR version 2016b (Windows, 64-bit) installed
- Microsoft Excel 2007 or later installed
- Recommended minimum disk space: 2 GB

For more information about "Standalone Applications" or "MATLAB Runtime", refers to MATLAB® Compiler™ document (MATLAB 2017).

INPUT DATASET

The *ISMLR application* is designed to solve time-series regression problems. Input dataset must be prepared in Excel format (*.xls or *.xlsx). The required dataset format (Figure 1) includes three elements: Headers, date/time, and data. The date/time information in the left-most column can appear in a variety of formats including "yyyy/m/d", "yyyy/m/d HH:MM", "yyyy/mm/dd", "yyyy/mm/dd HH:MM", or other similar formats. The response regressor appears in the far-right column; between date/time and response regressor are the columns of independent regressors. The first row contains headers or regressor names, and the remaining rows are date/time or data.



DATE	pH (t)	Water temperature (t)	NH_3-N (t-1)	SS (t-1)	Rawflow (t)	Precipitation (t)	Total flow (t+1)
2002/2/1	7.0	61	11.01	143.79	272.32	0.02	295
2002/2/2	7.4	61	10.18	134.55	235.05	0.00	238
2002/2/3	7.4	62	9.91	112.68	218.04	0.00	241
2002/2/4	7.3	62	9.26	80.43	211.86	0.00	226
2002/2/5	7.4	62	11.11	85.32	205.53	0.00	204
2002/2/6	7.3	62	14.24	119.29	204.00	0.00	222
2002/2/7	7.3	62			200.48	0.00	228
2002/2/8	7.2	61	15.13	130.06	214.27	0.00	253
2002/2/9	7.4	61	12.98	172.15	207.26	0.00	318
2002/2/10	7.5	61	9.03	125.99	233.38	0.07	290
2002/2/11	7.5	61	5.84	79.48	230.20	0.00	290
2002/2/12	7.4	62	8.16	91.34	229.01	0.00	269
2002/2/13	7.2	62	9.13	70.11	220.70	0.00	252
2002/2/14	7.4	61	11.16	85.85	211.27	0.00	234
2002/2/15	7.4	62	14.22	87.47	213.99	0.00	242
2002/2/16	7.6	62	14.26	92.92	206.31	0.00	224
2002/2/17	7.4	62	9.40	105.97	199.64	0.00	218
2002/2/18	7.3	62	9.85	185.39	204.19	0.00	266
2002/2/19	7.3	61	9.58	112.96	242.44	0.47	319
2002/2/20	7.3	61	12.64	228.21	273.41	0.21	304
2002/2/21	7.4	61	8.82	101.71	239.43	0.01	269
2002/2/22	7.3	62	9.79	66.30	224.64	0.00	254
2002/2/23	7.3	62	11.15	101.29	222.27	0.00	237
2002/2/24	7.4	62	9.13	83.49	205.88	0.00	229
2002/2/25	7.3	62	9.02	104.32	211.36	0.15	239
2002/2/26	7.3	62	11.08	95.99	218.13	0.12	225
2002/2/27	7.3	62	12.27	124.51	211.07	0.00	231
2002/2/28	7.3	61	12.12	93.03	208.17	0.00	224

Figure 1. An example input dataset, including date/time, independent regressors, and response regressor

The following guides and suggestions for preparing the datasets can help to facilitate a successful prediction:

- The two necessary datasets, training and testing, have to be prepared in a consistent format as described above.
- Categorical values will not work for the ISMLR application. Converting or transforming the categorical values to numerical data may solve the problem, but this approach has not been tested in detail.

- Proper management of outliers is critical. Common options for identifying and managing outliers often rely on standard deviation, Box plots, median value, or model-based methods. However, to avoid losing too much information, a simplified distance-based outlier detection heuristic method that was described by Zhu et al. (2015) and Zhu and Anderson (2016) can be used. The method can be briefly summarized as follows:
  1. Calculate the average value of each variable,
  2. Sort the data in an ordered list from the lowest to the highest values, and
  3. Calculate the differences between adjacent values in the ordered list.
  4. If there is a point in that list where a difference value exceeds the average value of the data, values from that point to the end of the list are considered to be outliers.

Examples of training and testing datasets are provided along with the ISMLR application, so users can use these datasets to be familiar with the application or create datasets based on their formats. Note that the data in the datasets were normalized.

SELECTING MODEL PARAMETERS

It is not necessary to specify model parameters, and default settings can be readily adopted. However, these parameters can be specified to achieve the different results. The first choice is to customize the *p*-value. A *p*-value quantifies the threshold probability associated with each variable in the regression model. Each of the selection steps is evaluated based on the change of *F*-ratios; the selection criteria, F_{enter} and F_{remove}, are defined using *p*-values for "enter" and "remove", respectively, which describe when a variable should be added or subtracted from a model. Default *p*-values are 0.05 and 0.10 for adding and subtracting, respectively, variables from the regression equation. Theoretically a *p*-value can be any value in the range from 0 to 1.0. In practice, assigning larger *p*-values may require a longer computation time but does not guarantee a better prediction performance (Zhu and Anderson 2017). The ISMLR application allows user to independently set *p*-values for the primary regression and subsequent regression(s). In addition, the *p*-value threshold for adding a variable to a model should always be less than a *p*-value for removing a variable from a model.

The second type of choice is to customize the regression function, which has four options:

- *Linear* (e.g. $y \sim x_1, x_2$),
- *Interactions* ($y \sim x_1, x_2, x_1{:}x_2$),
- *Purequadratic* ($y \sim x_1, x_2, x_1^2, x_2^2$), and
- *Quadratic* ($y \sim x_1, x_2, x_1{:}x_2, x_1^2, x_2^2$).

Relative to the other choices, *interactions* and *quadratic* usually take much more time to compute. As a result, unless there is a significant improvement in the performance of prediction, these two regression functions should be used with caution. Similar to the *p*-value, users can set different regression functions in the primary regression and the subsequent regression(s). More detailed information about *p*-values and regression can be found in documentation with MATLAB (2013) or in references such as the work by Montgomery and Runger (2010).

COMPUTATION TIME

The overall computation time will appear immediately after all computations are completed. Computation time could vary significantly depending on:

- Computer performance (CPU, RAM, hard drive, GPU)
- Computer resource usage (number and type of programs that are running simultaneously)
- Dataset size (number of observations and number of regressors), fraction of missing data, and fractions of training and testing data
- Training option selection, including *p*-values and regression types

Table 1 shows different computation times for predicting the next day's total flowrate, based on different combinations of respective regression types in primary and subsequent regressions (default *p*-values). An example of flowrate prediction using the ISMLR application will be described in a later section. Briefly, ten years of data (2002-2011) were divided into a training part (2002-2010) and a testing part (2011). The raw dataset includes 3652 observations, 105 independent regressors, and one response regressor. The computation time varies from less than one minute to more than 30 minutes. Because this example has a large number of regressors in the raw dataset, the regression types (such as *interactions* and *quadratic*) in the primary regression, can significantly increase the computation time. Because primary regression is mainly used to shrink the big cluster of regressors, it is usually a good idea to choose *linear* or *purequadratic* for that first step.

Table 1. An example of computation times (min) for flowrate prediction; computation times can vary a lot depending on the regression type in primary and subsequent regressions (j = 3).

	Subsequent regression(s)			
Primary regression	*Linear*	*Purequadratic*	*Interactions*	*Quadratic*
Linear	0.3	0.4	1.1	2.4
Purequadratic	0.6	0.6	1.9	2.0
Interactions	10.2	10.1	10.5	12.1
Quadratic	29.0	27.4	29.3	30.9

*Test computer specifications:

CPU:	Intel® Core™ i5-2520 M (2.50 GHz)
RAM:	8.00 GB, 1333MHz DDR3
GPU:	NVIDIA NVS 4200M
Hard drive:	500 GB, 5400 rpm
OS:	Windows 7, SP 1 (64-bit)

OUTPUT INFORMATION AND DOCUMENTS

ISMLR prediction results will be automatically presented as figures and tables in the ISMLR application interface, and the results be exported to individual Excel spreadsheets. The main interface of the ISMLR application presents a figure showing the predicted values of the response regressor as a function of the measured values, including the training dataset and the testing dataset. Additional output information, which can be accessed using buttons on the interface, includes ordinary time-series predictions, residuals, and pre-/post-ISMLR:

- **Time-series**. Measured data of response regressor are plotted in time-series based on the final model (treated datasets, no missing data), their corresponding predicted values are also shown in the same figure. A figure is plotted for each of the two (training and testing) datasets.
- **Residual**. Residual values (differences between measured and predicted values) are plotted with their corresponding predicted values. Here again there are two figures, one for each dataset.
- **ISMLR (training)**. Similar to the time-series plots, but these are based on training datasets that include the days with missing data. The *Pre-ISMLR* is the model/plot based on an initial raw dataset, whereas the *Post-ISMLR* is the model/plot based on the final dataset that only includes important regressors.
- **ISMLR (testing)**. Similar to the ISMLR (training), but this button shows the results based on the testing datasets.

In addition to the above plots, three major tables are presented in the main interface. They summarize important regressors, iterations, and prediction performance:

- **Subset of important regressors**. The final important regressors are summarized in this table. When other types of regression other than *linear* are selected, second order regressors may be found in the list and they are listed as “A × B” or “A × A” where “A” and “B” represent different individual regressors. The regressors and their coefficients are listed in the order of importance based on their *p*-values.
- **Iteration summary**. This table summarizes the number of iterations, the number of individual regressors, the number of all regressors, the number of observations, and the retention level (%). The number of iterations is the number of times that a treated dataset is built; the number of individual regressors accounts for individual independent regressors; the number of all regressors accounts for all linear, interaction, and pure-quadratic regressors; the number of observations accounts for all valid time-series rows (months, days, hours, or minutes) that include both training part and testing part; the retention level expresses the valid number of observations as a percentage of the total number of observations. Note that the number of observations can change during the modeling process because the initial analysis only includes the training part but the final number accounts for both training part and testing part.
- **Prediction performance**. The performance of the pre-ISMLR and post-ISMLR is evaluated based on five criteria: R^2, adjusted R^2, root mean of squared error (RMSE), mean relative error (MRE, %), and mean absolute error (MAE). These parameters are defined as shown in the following equations:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

$$Adjusted\ R^2 = 1 - \left[\frac{(1 - R^2)(n - 1)}{n - k - 1}\right]$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$$

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

All the above results and data are automatically exported to five Excel spreadsheets, readily available for further use and research. The five Excel documents are:

- **Processed training dataset**. This spreadsheet includes the final treated training dataset (final important regressors and cleaned observations) as well as predicted values of the response regressor and their corresponding residual values.
- **Processed testing dataset**. Similar information as for the processed training dataset, but applied to the testing dataset.
- **Evaluation summary**. This spreadsheet includes the prediction performance summary, subset of important regressors, computation time, user-selected modeling options, and iteration summary.
- **Pre-&Post-ISMLR training part**. This spreadsheet includes time-series measured data and predicted values based on the pre-ISMLR model and the post-ISMLR model for the training part.
- **Pre-&Post-ISMLR testing part**. Similar information as for the above training part, but applied to the testing part.

EXAMPLE 1

Objective: Predict the next day's total flow, $Q_t\ (t+1)$, at the MWRDGC Calumet WRP

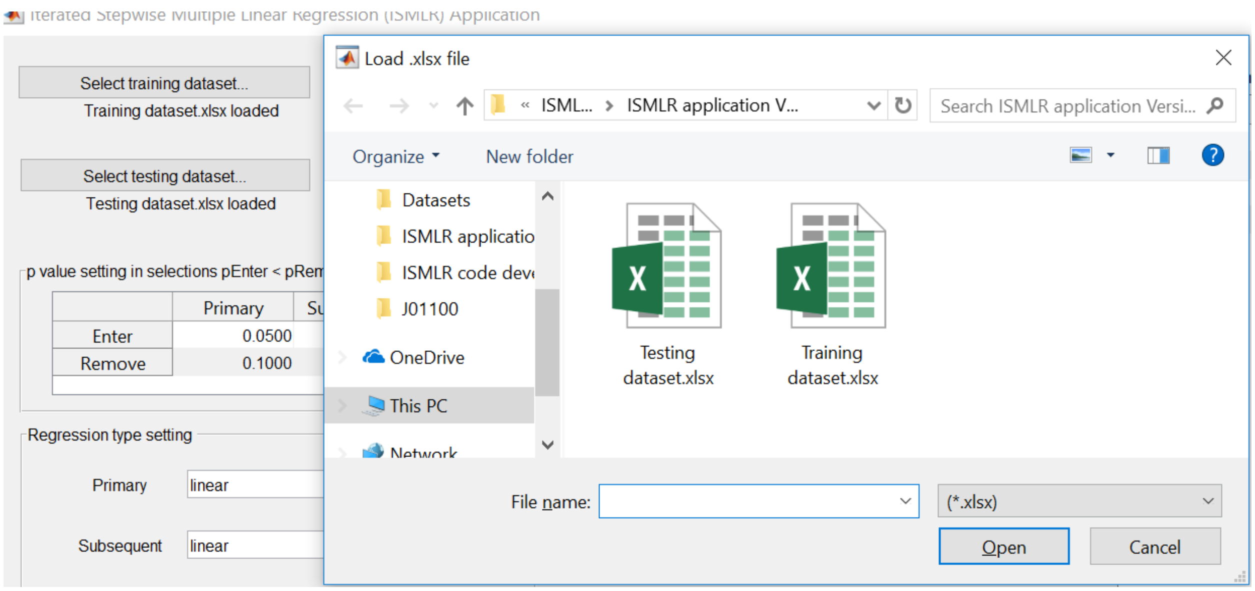
Datasets: Ten years (2002-2011) of historical data at the MWRDGC Calumet WRP, the first nine years of data were the training dataset and the last one year of data was the testing dataset.

Independent variables: 14 variables were used to develop 105 regressors, including 98 historical regressors, five "real-time" (the current day) regressors, and two "future" (the next day) regressors.

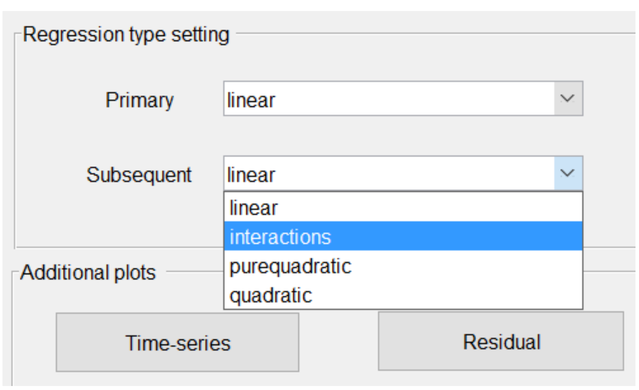
Regression options: Default *p*-values; primary regression type: *Linear*; subsequent regression type: *Interactions*.

Working procedures

1. Load datasets. The first step is to load the training and testing datasets.



2. Option setting. Choose *interactions* in the subsequent regressions.



3. Run the application. Start the computation by clicking the button "Run ISMLR…".

When the program is running, check the status of computation in the bottom, left corner:

Status: Primary iteration is calculating......
Primary iteration is calculating......

Status: Iteration 2 is calculating......
Primary iteration has been completed (9.386 seconds used)

Status: Iteration 3 or confirmation step is calculating......
Iteration 2 has been completed (23.955 seconds used)

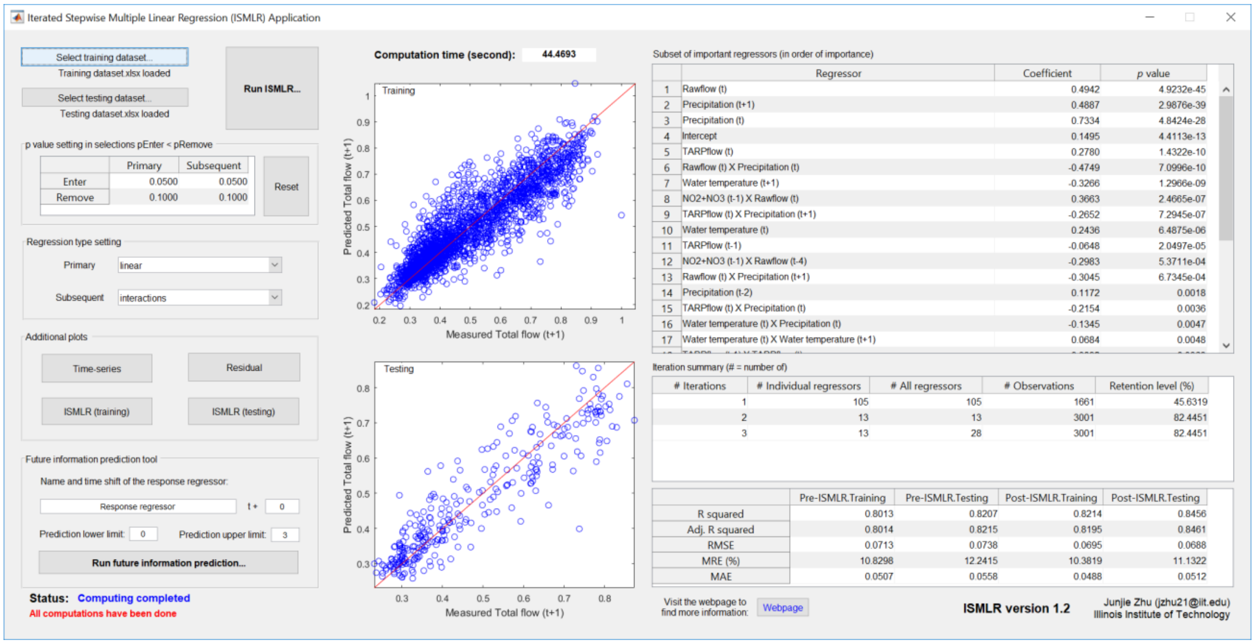
Status: Summarizing, plotting, and exporting data......
Main function has been completed (38.417 seconds used)

Status: Computing completed
All computations have been done

The iteration summary and prediction performance are updated immediately after each iteration.

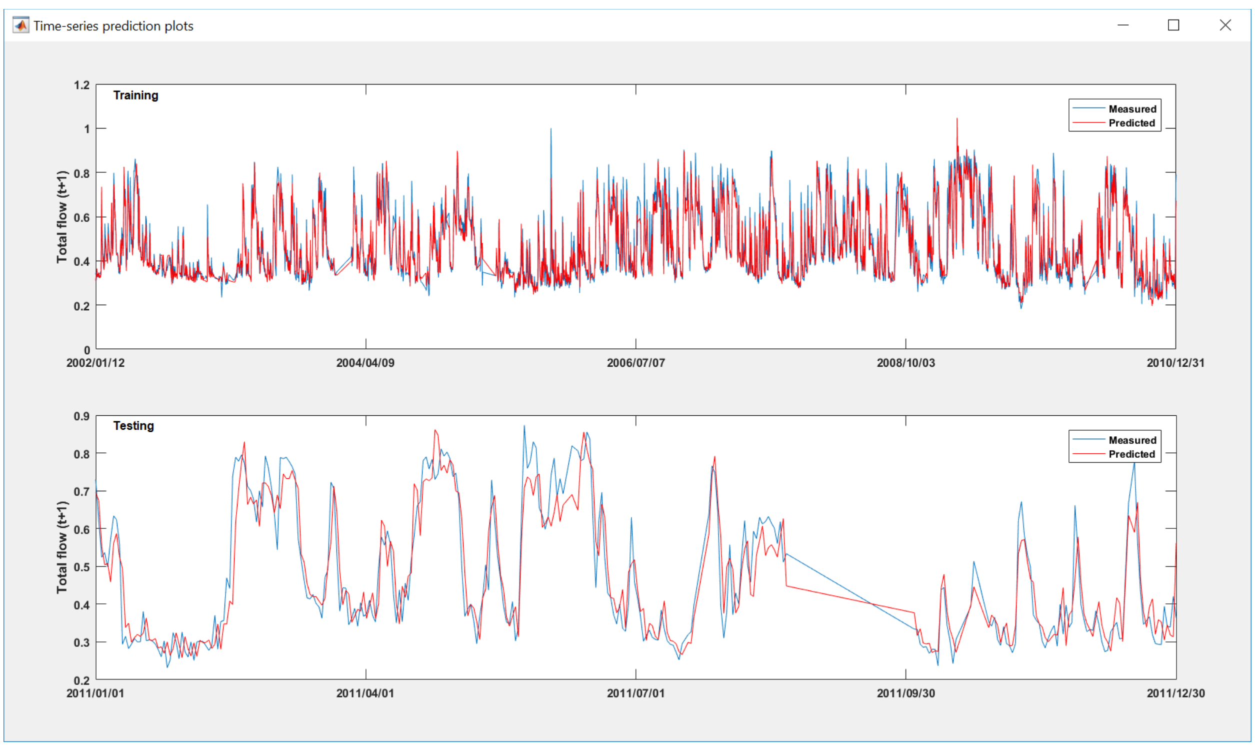
Computation results

1. Main interface

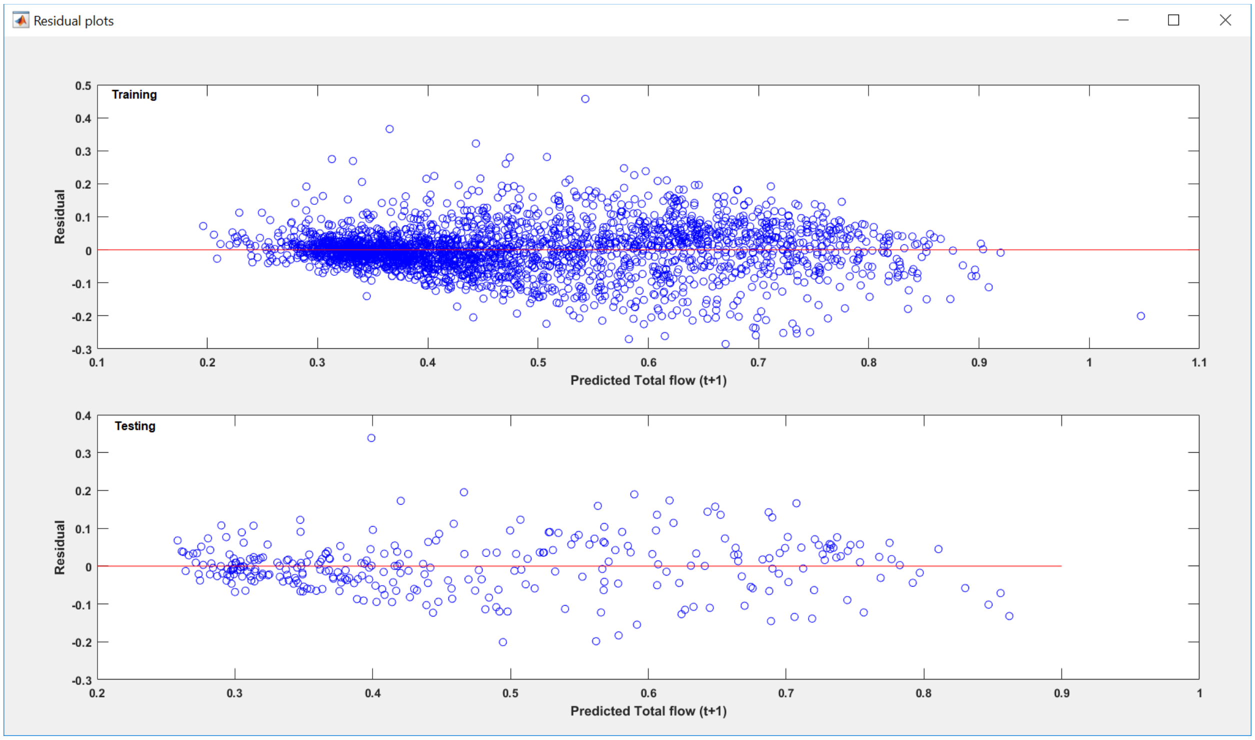


2. Time-series plots

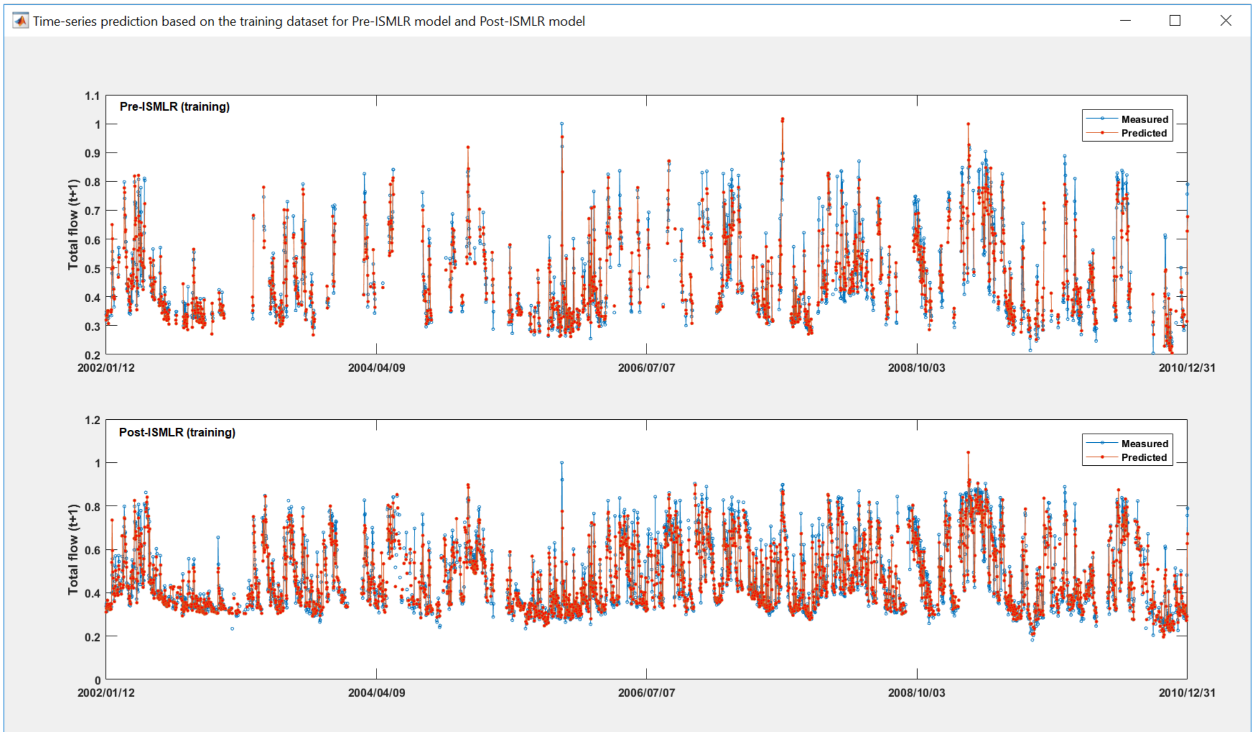
The x-axis of all time-series plots is equally divided into four ranges and given by five corresponding time labels (for example: yyyy/mm/dd).



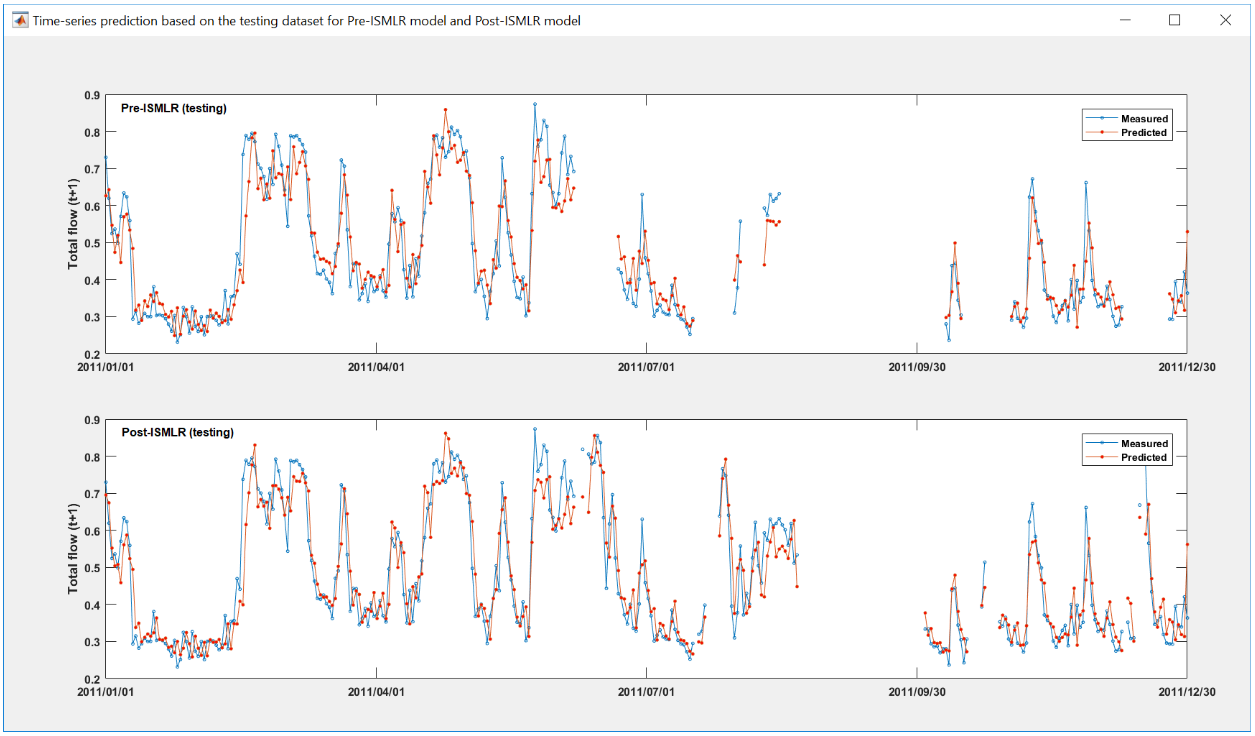
3. Residual plots



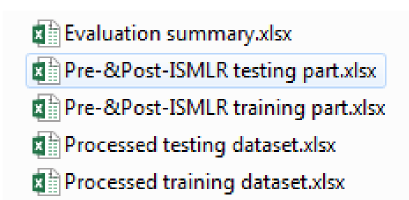
4. Pre-/Post-ISMLR (training) plots



5. Pre-/Post-ISMLR (testing) plots

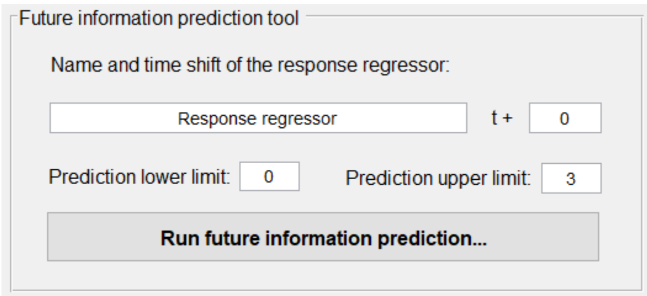


6. Output of excel spreadsheet documents.



FUTURE INFORMATION PREDICTION TOOL

A future information prediction (FIP) tool, developed to obtain future information of the response variable ahead for a period, has been included in the ISMLR application since version 1.2. The FIP works similar to the original ISMLR modeling that has options of *p*-values and regression types, and the user-defined modeling options will be applied to predict all future variables. Additional options and inputs are available in the FIP tool:



Two required inputs are the name and time shift of the response regressor. The time shift means the period that the response regressor shifts from the current time. For example, it will be "*t+0*" if we are going to predict the current day's influent flowrate (at a WRP) in the original input dataset, whereas it will be "*t+1*" if the response regressor is the next day's influent ammonia concentration.

Two options are prediction lower limit (PLL) and prediction upper limit (PUL), which are used to set the period of the response regressor that the user wants to predict. For example, assume that we are going to predict the next three days' (*t+1* ~ *t+3*) influent flowrate while the response regressor in the original dataset is $Q_t\ (t+1)$ (example 1). The PLL and PUL should be set to 0 (t + 1 + 0 = t + 1) and 2 (t + 1 + 2 = t + 3), respectively.

EXAMPLE 2

Objective: Predict the next seven day's ultraviolet absorbance (UVA), *UVA (t+1)* ~ *UVA (t+7)*, in the Illinois Fox River

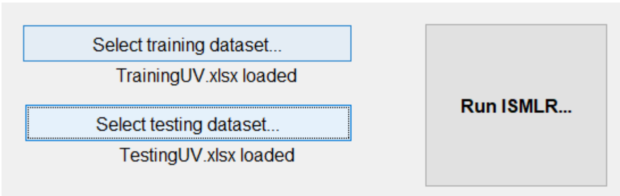
Datasets: Eight years (2002-2009) of historical data at or close to City of Elgin, the first six years of data were the training dataset and the last two years of data were the testing dataset. Some information can be found in Zhu (2012)

Independent variables: Five variables were used to develop 28 regressors, including 22 historical regressors, five "real-time" (the current day) regressors, and one "future" (the next day) regressor.

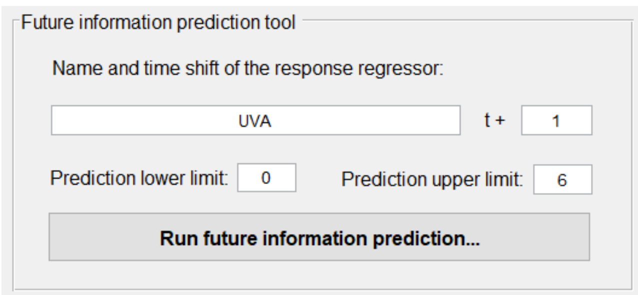
Regression options: Default *p*-values and regression type (*Linear* + *Linear*).

Working procedures

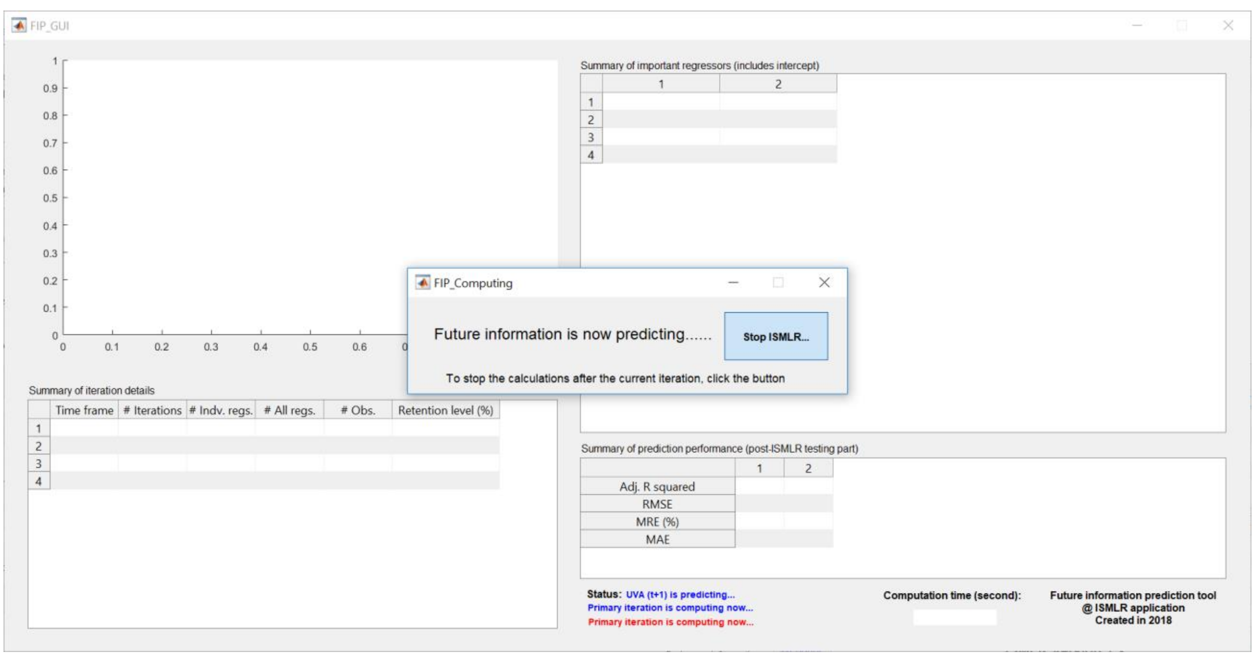
1. Load datasets. The first step is to load the training and testing datasets. In the dataset, the response regressor is *UVA(t+1)*.



2. Option setting and additional information. The existing response regressor in input datasets is UVA (t+1), and PLL and PUL are set to 0 and 6, respectively.



3. Run the FIP tool. Start the computation by clicking the button "Run future information prediction…". A new window will be popped up and prediction results will be updated and shown in the tables iteratively.



When the program is running, check the status of computation in the bottom:

Status: UVA (t+1) is predicting...
Primary iteration is computing now...
Primary iteration is computing now...

Status: UVA (t+1) is predicting...
Iteration 2 is calculating......
Primary iteration has been completed (1.063 seconds used)

Status: UVA (t+1) is predicting...
Iteration 3 or confirmation step is calculating......
Iteration 2 has been completed (1.288 seconds used)

Status: UVA (t+1) is predicting...
Summarizing and exporting data......
Main function has been completed (1.497 seconds used)

……

Status: UVA (t+4) is predicting...
Primary iteration is computing now...
Primary iteration is computing now...

Status: UVA (t+4) is predicting...
Iteration 2 is calculating......
Primary iteration has been completed (26.858 seconds used)

Status: UVA (t+4) is predicting...
Summarizing and exporting data......
Main function has been completed (27.448 seconds used)

……

Status: UVA (t+7) is predicting...
Primary iteration is computing now...
Primary iteration is computing now...

Status: UVA (t+7) is predicting...
Iteration 2 is calculating......
Primary iteration has been completed (55.212 seconds used)

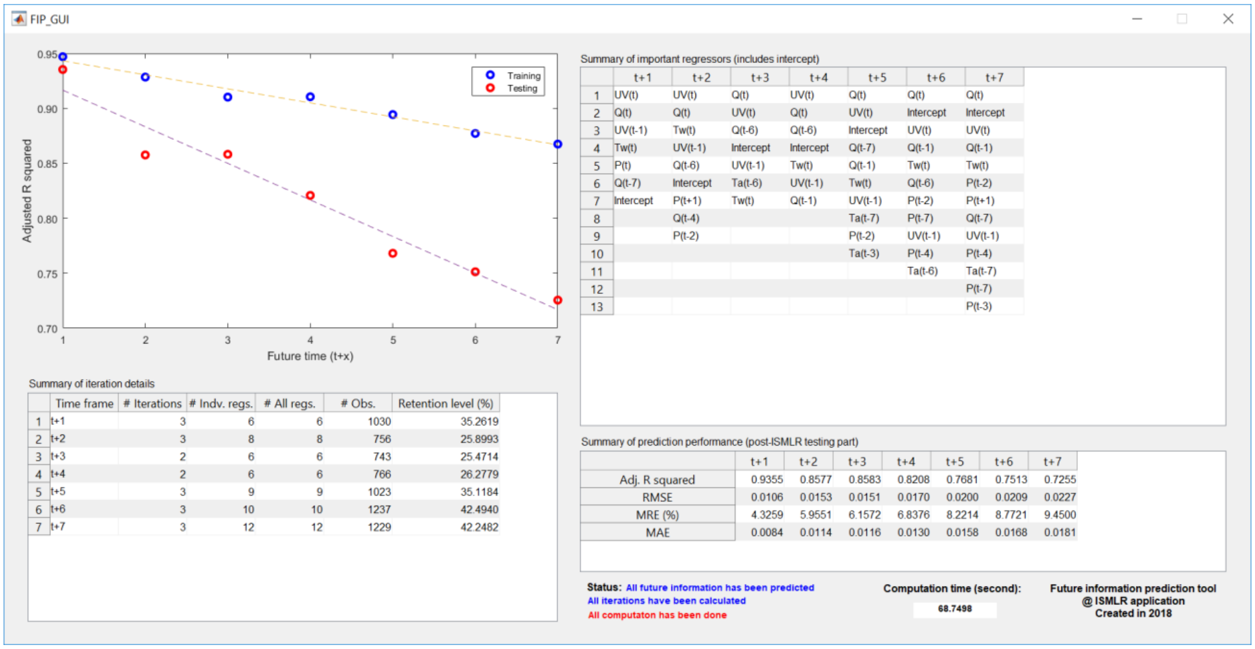
Status: UVA (t+7) is predicting...
Summarizing and exporting data......
Main function has been completed (58.288 seconds used)

Status: UVA (t+7) is predicting...
This iteration has been done
This iteration has been done

Status: All future information has been predicted
All iterations have been calculated
All computaton has been done

Computation results

1. Main results interface of the FIP tool



In the results interface of the FIP tool, a plot shows the adjusted R^2 values for predicting the response regressor from one day ahead (*t+1*) to seven days ahead (*t+7*) based on training dataset and testing dataset. In addition to the plot, three tables are shown the details of models and specific predictive performance. For example, for *UVA (t+3)*, it has two iterations, the final model includes six individual/all regressors. 743 observations are retained with a retentional level of 25.5%. The most important regressor is *Q (t)*. Adj. R^2, RMSE, MRE, and MAE are 0.858, 0.0151 (1/cm), 6.157%, and 0.0116 (1/cm), respectively.

2. Output of excel spreadsheet documents.

- Evaluation summary(FIP).xlsx
- Processed testing dataset(FIP).xlsx
- Processed training dataset(FIP).xlsx
- Pre-&Post-ISMLR testing part(FIP).xlsx
- Pre-&Post-ISMLR training part(FIP).xlsx

TIPS

The ISMLR application needs to use the resources of MATLAB Compiler Runtime (MCR), so the initial loading of the program may take longer than expected. You may also need to wait if:

- A large dataset is being loaded
- Regression type *interactions* or *quadratic* is applied
- System performance is relatively low

The ISMLR application can**not** work if:

- Input training and testing datasets are not consistent
- Row(s) in the dataset(s) include data and but not the corresponding date/time

In addition, although it is rare, the ISMLR application cannot work if there are no data missing in the raw datasets and all the regressors are important, because the ISMLR application will fall into the infinite loops of SMLR. In this case, you can manually add one additional dummy column, giving “1” to all the values and proceed.

Please check all the above items before running the application, so you can avoid these common issues. During computation, you can stop the application by click the “Stop ISMLR” button; the program cannot be stopped during an iteration, but it will stop after the current iteration. Alternatively, you can simply close and restart the ISMLR application. Please send email to the author (jzhu21@iit.edu) if you discover any other problems.

USE AND DISTRIBUTION

Use of the *ISMLR application* for any commercial purpose is prohibited. The ISMLR application is designed for non-profit activities (teaching and research) and it can be downloaded for free for these uses. The author encourages users to spread the program, share their user experiences, and make comments and suggestions. All these steps will improve the ISMLR application.

If you have publications that include results from using the ISMLR application, please include a proper citation. To cite the conventional ISMLR method, use Zhu and Anderson (2016); to cite the peer-viewed paper about ISMLR application, use Zhu and Anderson (2018) (in preparation); to cite the program of ISMLR application or this instruction, use Zhu (2018).

Finally, here are some other suggestions:

- To download and follow up recent updates of the *ISMLR application*, please visit the author's webpage, Researchgate, or GitHub.
- You can make comments and suggestions using the author's webpage, Researchgate, or via email.
- Any discussion with the author in terms of potential collaboration via email is welcome.

Personal webpage: https://junjiezhublog.wordpress.com/
Researchgate: http://www.researchgate.net/profile/Junjie_Zhu4
GitHub: https://github.com/starfriend10/ISMLR-application
Email address: jzhu21@iit.edu

We are also planning to develop ISMLR code using powerful open source software, such as Python or R.

ACKNOWLEDGEMENT

During the development of ISMLR application, I had many discussions with my former advisor, Paul R. Anderson. Paul provided many useful suggestions on the interface designing and language presentation. Robert Nunoo and Boyang Lu are Ph.D. candidates in Anderson's research group; thank for their helps in testing the ISMLR application using their computers.

For the Calumet WRP datasets used in the example 1, I wish to thank Dr. Catherine O'Connor and Judith Moran, Metropolitan Water Reclamation District of Greater Chicago; for providing the data. For the Illinois Fox River datasets used in the example 2, I wish to thank Kyla Jacobsen, Riverside Water Treatment Facility at City of Elgin; for providing the data.

REFERENCES

MATLAB. (2013). Function "*stepwiselm*", introduced in version R2013b, is used to create linear regression model using stepwise regression. https://www.mathworks.com/help/stats/stepwiselm.html (accessed March 22, 2018)

MATLAB. (2017). Document of MATLAB® Compiler™. https://www.mathworks.com/help/compiler/ (March 22, 2018)

Montgomery, D. C., & Runger, G. C., (2010). *Applied statistics and probability for engineers*. Fifth Edition. Published by John Wiley & Sons, Inc. ISBN: 9780471204541

Zhu, J.-J. (2012). *Investigation on the interaction between natural organic matter and calcite*. M.S. Thesis. Illinois Institute of Technology, Chicago, IL. [Official webpage] [Document shared]

Zhu, J.-J., Segovia, J., & Anderson, P. R. (2015). Defining influent scenarios: Application of cluster analysis to a water reclamation plant. *J. Environ. Eng*., DOI: 10.1061/(ASCE)EE.1943-7870.0000934. [Official webpage] [Document shared]

Zhu, J.-J. (2015). *Cyber-physical system for a water reclamation plant: Balancing aeration, energy, and water quality to maintain process resilience*. Ph.D. Dissertation. Illinois Institute of Technology, Chicago, IL. ProQuest/UMI. Publication Number: AAT 3733990; ISBN: 9781339224329. [Official webpage] [Document shared]

Zhu, J.-J. (2018). ISMLR application, version 1.0 was published on August 01, 2017. Its latest version 1.2 (published on March 22, 2018) and instructions can be downloaded from the following sources:
Personal webpage https://junjiezhublog.wordpress.com/,
Researchgate http://www.researchgate.net/profile/Junjie_Zhu4, or
GitHub https://github.com/starfriend10/ISMLR-application (accessed March 22, 2018)

Zhu, J.-J., & Anderson, P. R. (2016). Assessment of a soft sensor approach for determining influent conditions at the MWRDGC Calumet WRP. *J. Environ. Eng*. DOI: 10.1061/(ASCE)EE.1943-7870.0001097. [Official webpage] [Document shared]

Zhu, J.-J., & Anderson, P. R. (2018). ISMLR: A MATLAB-based application for managing missing data and predicting time-series observations. In preparation.

Zhu, J.-J., Kang, L., & Anderson, P. R. (2018). Predicting influent biochemical oxygen demand: Balancing energy demand and risk management. *Water Res*., 128(C), 304-313. DOI: 10.1016/j.watres.2017.10.053. [Official webpage] [Document shared]